PCS 2428 / PCS 2059
Inteligência Artificial

Prof. Dr. Jaime Simão Sichman
Prof. Dra. Anna Helena Reali Costa

Satisfação de Restrições

Constraint Satisfaction Problems (CSP)

- Conceitos básicos
- Busca cega simples e refinada
- Busca heurística

2

Constraint Satisfaction Problems (CSP)

- Um Problema de Satisfação de Restrições
  - tipo de problema que impõe propriedades estruturais adicionais à solução a ser encontrada
  - há uma demanda mais refinada do que na busca clássica
    - ex. ir de Recife à Cajazeiras com no máximo 3 tanques de gasolina e 7 horas de viagem
- Um CSP consiste em:
  1. um conjunto de variáveis que podem assumir valores dentro de um dado domínio
  2. um conjunto de restrições que especificam propriedades da solução - <u>valores que essas variáveis *podem* assumir.</u>

3

Constraint Satisfaction Problems (CSP)

- Formulação
  - **Estados:** definidos pelos valores possíveis das variáveis
  - **Estado inicial:** nenhuma variável instanciada ainda
  - **Operadores:** atribuem valores (instanciação) às variáveis
    - uma variável por vez
  - **Teste de término:** verificar se todas as variáveis estão instanciadas obedecendo as restrições do problema
  - **Solução:** conjunto dos valores das variáveis instanciadas
  - **Custo de caminho:** número de passos de atribuição

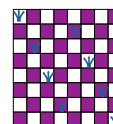
4

CSP: características das restrições

- O conjunto de valores que a variável *i* pode assumir é chamado domínio *Di*
  - O domínio pode ser **discreto** (fabricantes de uma peça do carro) ou **contínuo** (peso das peças do carro)
- Quanto à aridade, as restrições podem ser
  - unárias (sobre uma única variável)
  - binárias (sobre duas variáveis)
  - n-árias
    - a restrição unária é um sub-conjunto do domínio, enquanto que a n-ária é um produto cartesiano dos domínios
- Quanto à natureza, as restrições podem ser
  - **absolutas** (não podem ser violadas)
  - **preferenciais** (devem ser satisfeitas quando possível)

5

Exemplo

- Problema das 8-rainhas
  - *variáveis*: localização das rainhas (coluna, linha)
  - *valores*: possíveis posições do tabuleiro
  - *restrição binária*: duas rainhas não podem estar na mesma coluna, linha ou diagonal
  - *solução*: valores para os quais a restrição é satisfeita

6

Busca cega com Retrocesso para CSP

- Funcionamento
  - estado inicial: variáveis sem atribuição
  - aplica operador: instancia **uma** variável
  - teste de parada: todas variáveis instanciadas sem violações
- Retrocesso (*Backtracking*)
  - depois de realizar uma atribuição, verifica se restrições não são violadas
  - caso haja violação ⇒ retrocede
- Análise
  - pode ser **busca em profundidade limitada** (l = número de variáveis)
  - é completa
  - fator de expansão: $\sum_i |Di|$
  - o teste de parada é decomposto em um conjunto de restrições sobre as variáveis

7

Backtracking não basta...

- Problema do *backtracking*:
  - não adianta mexer na 7a. rainha para poder posicionar a última
  - O problema é mais em cima... O *backtrack* normalmente tem que ser de mais de um passo
- Soluções:
  - Verificação prévia *(forward checking)*
  - Propagação de restrições

15

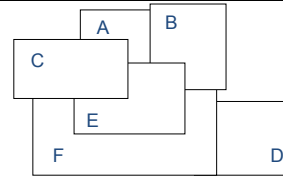
Verificação Prévia

- Verificação prévia (forward checking)
  - **idéia**: olhar para frente para detectar situações insolúveis
- Algoritmo:
  - Após cada atribuição, **elimina do domínio** das variáveis não instanciadas os valores incompatíveis com as atribuições feitas até agora.
  - Se um domínio torna-se vazio, retrocede imediatamente.
- É bem mais eficiente!

16

Exemplo: coloração de mapas

variáveis: A,B,C,D,E,F
domínio: Da=Db...=Df={green, red, blue}
restrições: A ≠ B; A ≠ C; A ≠ E; B ≠ E; B ≠ F; C ≠ E; C ≠ F; D ≠ F; E ≠ F

A B C E F D

Solucionar usando busca em profundidade limitada com l=6 e verificação prévia.

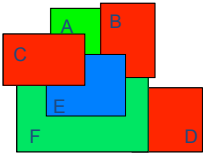
17

Propagação de restrições

- Propagação de restrições (constraint propagation)
  - uma conseqüência da verificação prévia
  - quando um valor é eliminado, isto é **propagado** para outros valores que dele dependem, podendo torná-los inconsistentes e eliminados também
  - é como uma onda que se propaga: as escolhas ficam cada vez mais restritas

19

Heurísticas para CSP

- Tenta reduzir o fator de expansão do espaço de estados
- Onde pode entrar uma heurística?
  - Ordenando a escolha da **variável** a instanciar
  - Ordenando a escolha do **valor** a ser associado a uma variável
- Existem 3 heurísticas para isto...
  - ***Variável mais restritiva***: variável envolvida no maior número de restrições é preferida
  - ***Variável mais restringida***: variável que pode assumir menos valores é preferida
  - ***Valor menos restritivo***: valor que deixa mais liberdade para futuras escolhas

22

Variável mais restritiva

(variável envolvida no maior número de restrições)

variáveis: A,B,C,D,E,F
domínio: Da=Db...=Df={green,red, blue}
restrições: A ≠ B; A ≠ C; A ≠ E; B ≠ E; B ≠ F; C ≠ E; C ≠ F; D ≠ F; E ≠ F

Candidatas[1]: E, F, ...resto
E = green
Candidatas: F, ...resto
F = red
Candidatos: A, B, C, D
A= red
Candidatos: B, C, D
B= blue
Candidatos: C, D
C= blue
D = green

SEM BACKTRACK!!

A B C E F D

23

[1] *em ordem de prioridade (4, 4, 3, 3, 3, 1)*

CSP – conclusões

- Grande importância prática, sobretudo em tarefas de
  - Criação, projeto (design)
  - Agendamento (scheduling)
  - onde várias soluções existem e é mais fácil dizer o que não se quer...
- Estado atual
  - Grandes aplicações industriais $$$$
  - Número crescente de artigos nas principais conferências
- Observação:
  - a sigla CSP também é usada para falar de ***Constraint Satisfaction Programmimg***, que é um paradigma de programação

26